**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ**

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ**

**«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук

Департамент программной инженерии



Документация по командному проекту

«Программный комплекс для обработки больших астрономических данных»

Студентов 4 курса Департамента ПИ

Долгушева Андрея Васильевича,

Лукина Ильи Владимировича,

Соловьева Егора Александровича

|  |  |  |
| --- | --- | --- |
| Заказчик | Личная подпись | \_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_ |
| Ответственный по проекту | Личная подпись | \_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_ |

Дата:

**СОДЕРЖАНИЕ**

Оглавление

[1. Введение 2](#_Toc3320107)

[1.1 Цель и назначение документа 2](#_Toc3320108)

[1.2 Целевая аудитория документа 2](#_Toc3320109)

[1.3 Наименование программы 2](#_Toc3320110)

[1.4 Краткая характеристика области применения программы 2](#_Toc3320111)

[1.5 Функциональное и эксплуатационное назначение 2](#_Toc3320112)

[2. Проектная документация 3](#_Toc3320113)

[2.1. Требования к функциональным характеристикам 3](#_Toc3320114)

[2.2. Требования к интерфейсу 3](#_Toc3320115)

[2.3. Требования к надежности и условия эксплуатации 3](#_Toc3320116)

[2.4. Условия эксплуатации 3](#_Toc3320117)

[2.5. Описание конкурентов 4](#_Toc3320118)

[3. Спецификации к программному обеспечению 4](#_Toc3320120)

[3.1. Ключевые файлы приложения 4](#_Toc3320121)

[3.2. Функции и подпрограммы 4](#_Toc3320122)

[3.3. Переменные и константы программы 4](#_Toc3320124)

[3.4. Описание кода (интерфейсы, классы). 5](#_Toc3320125)

[3.5. Архитектура решения 5](#_Toc3320126)

[3.6. План тестирования и баг-репорты 5](#_Toc3320127)

[4. Руководство пользователя 5](#_Toc3320128)

[4.1. Пояснения к действиям пользователя для выполнения функций 6](#_Toc3320129)

[4.2. Пояснения к экранным формам 6](#_Toc3320130)

[4.3. Пояснения к выходным данным, способы интерпретирования выходных данных 6](#_Toc3320131)

[4.4. Пояснения к возможным ошибкам 6](#_Toc3320132)

1. Введение
   1. Цель и назначение документа

Цель данного документа – предоставить информацию о следующих документах:

* Архитектурная/проектная документация — обзор программного обеспечения, включающий описание рабочей среды и принципов, которые должны быть использованы при создании ПО
* техническая документация — документация на код, алгоритмы, интерфейсы, API
* пользовательская документация - руководства для конечных пользователей, администраторов системы и другого персонала
  1. Целевая аудитория документа

Целевой аудиторией документа являются:

· Заказчик проекта

· Куратор проекта

· Команда разработчиков

· Конечные пользователи

* 1. Наименование программы

Наименование программы: «prao-compresser».

* 1. Краткая характеристика области применения программы

Данная программа может быть использована с целью первичной обработки получаемых данных с Большой синфазной антенны (БСА) Пущинской радиоастрономической обсерватории. Обработка включает в себя калибровку, сжатие и визуализацию сжатых данных. Данная программа обеспечивает более эффективную работу Обсерватории над ее главной задачей – поиском пульсаров и других объектов глубокого космоса.

* 1. Функциональное и эксплуатационное назначение

Программный продукт является инструментом работы с Big Data в сфере высокопроизводительных вычислений.

1. Проектная документация

**<описание требований к продукту, ограничений, показателей качества>**

* 1. Требования к функциональным характеристикам

1. Возможность калибровки данных с помощью калибровочных сигналов (с учетом влияния погодных условий)
2. Возможность потоковой обработки данных для всех лучей, всех частот с выделением для на каждом отрезке времени в несколько секунд: среднемедианных значений, стандартных уклонений данных, максимальных, минимальных значений и т.п.
3. Возможность внесение полученных значений в базу данных либо в бинарные сжатые файлы значений
4. Возможность статистического анализа качества данных
5. Возможность выделения в обработанных данных радиоисточников и анализа кратковременных и долговременных изменений потоков от них
6. Возможность кросс-анализа данных с целью выявления вспышечных явлений от источников космической природы (сигнал приходит вначале на высокие частоты, на низкие частоты приходит с задержкой)
7. Возможность использование математических фильтров для выделения слабых сигналов известной формы
   1. Требования к интерфейсу

Программный комплекс должен иметь минимальный консольный интерфейс для ввода параметров работы программы. Программный комплекс должен иметь графический интерфейс пользователя для показа графиков, построенных на основе сжатых данных.

* 1. Требования к надежности и условия эксплуатации
  + Практичность (Usability)

ПК должен предоставлять возможность получения и обработки данных, поступающих с телескопа. Основной целью является обнаружение новых космических объектов.

* + Надежность

ПК должен функционировать бесперебойно при условии наличии источников питания и отсутствия сбоев в окружающей ОС.

* + Производительность (Performance)

ПК должен обрабатывать поступающие данные в режиме реального времени. Допускается задержка в 10с.

* + Возможность обслуживания (Supportability)

ПК должен быть спроектирован таким образом, что добавление нового функционала или исправление существующих дефектов не вызывало существенных изменений в архитектуре ПК.

## Условия эксплуатации

## Описание конкурентов

Конкурентов нет.

1. Спецификации к программному обеспечению

**< определение и описание**[**API**](https://ru.wikipedia.org/wiki/API)**,**[**структур данных**](https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85)**и**[**алгоритмов**](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC)**>**

* 1. **Ключевые файлы приложения**

Compresser.h – главный класс, координирующий калибровку, подсчет метрик, сжатие, запись в файл.

OpenCLContext.h – класс, хранящий информацию о вычислительном устройстве.

DataReader.h – считыватель данных

CalbrationDataInput.h – калибровщик данных

MetricsCalclulator.h – класс, подсчитывающий статистические метрики

* 1. **Функции и подпрограммы**

Программа состоит из двух подпрограмм.

1. Подпрограмма, осуществляющая обработку поступающих данных. Обладает следующими функциями: считывание данных, калибровка, сжатие на основе подсчета статистических метрик, запись в файл. Работа данной подпрограммы являет собой последовательное выполнение всех вышеперечисленных функций, у пользователя нет возможности выбрать какую-либо отдельную функцию для работы.
2. Подпрограмма, осуществляющая визуализацию сжатых данных. Визуализация осуществляется посредством построения графиков подсчитанных статистических метрик в разрезе времени.
   1. **Переменные и константы программы**

device – используемое устройство

algorithm – используемый алгоритм

leftPercentile – левый перцентиль

rightPercentile – правый перцентиль

starSeconds – коэффициент сжатия

localWorkSize – размер локальной work group

fileListPath – путь к файлу содержащим пути к файлам, содержащим данные

calibrationListPath – путь к файлу, содержащим данные калибровки

outputDirectory – место записи выходных данных

* 1. **Описание кода (интерфейсы, классы).**

1. Модуль Calibration – хранит в себе классы, используемые для

* CalibrationDataInput.h – класс используемый для
* CalibrationData.h
* CalibrationStorage.h

1. Модуль Metrics – хранит информацию о метриках

* MetricsType.h – класс, содержащий в себе значения метрик
* MetricsContainer.h – класс, используемый для записи выходных данных в файл

1. Модуль Point

* Point.h – содержит информацию о точке из входного файла

1. Модуль Processing – содержит в себе классы, совершающие основные подсчеты

* Processor.h – родительский класс, содержащий метод run() и хранящий контекст вычисляющего устройства
* MetricsCalculator.h – наследник класса Processor, совершает подсчеты метрик
* OpenCLContext.h – контекст вычисляющего устройства. Создается один раз.
* Пакет Kernels содержит в себе ядра, код которых выполняется на вычисляющем устройстве.

1. Модуль Reader – содержит в себе классы, используемые при чтении данных

* DataReader.h – класс, реализующий чтение данных
* FileListItem.h – класс, представляющий единичный файл с данными

1. Модуль Time

* Time.h – класс, используемый для правильного подсчета времени
  1. **Архитектура решения**

Главным архитектурным решением программы является использование паттерна проектирования Chain of Responsibility. Входные данные последовательно обрабатываются модулями подпрограммы, преобразуя входные данные в выходные. Также для выбора применяемого алгоритма используется паттерн Стратегия. Класс Processor.h является реализацией паттерна Шаблонный метод. OpenCLContext является Синглтон-классом, он создается один раз и передается внутрь наследника класса Processor. Также активно применялись наследование, агрегация и композиция.

* 1. **План тестирования и баг-репорты**

**…**

1. Руководство пользователя

**<описание функций программы и принципов их использования. Алгоритм действий пользователя в случае возникновения проблем>**

* 1. Пояснения к действиям пользователя для выполнения функций

Для запуска работы программы достаточно запустить ее через консоль ОС с аргументом, содержащим путь к конфигурационному файлу, который в свою очередь содержит значения необходимых входных параметров. Программа не предполагает выбор функций.

* 1. Пояснения к экранным формам

Экранные формы отсутствуют, взаимодействие с пользователем осуществляется через консоль ОС. Пояснения к работе с модулем визулизации будут даны в следующем пункте.

* 1. Пояснения к выходным данным, способы интерпретирования выходных данных

**Егор жги**

* 1. **Пояснения к возможным ошибкам**

При возникновении ошибки программа выведет в консоль соответствующую информацию и прекратит работу. Также возможен вывод кода ошибки, их интерпретацию можно найти по следующей ссылке: https://streamhpc.com/blog/2013-04-28/opencl-error-codes/

Дополнительные источники, которые могут быть полезны при разработке документации:

<https://www.inflectra.com/ideas/topic/requirements-definition.aspx>

<https://habr.com/ru/post/52681/>